# Global optimization software library for research and education

*Nadia Udler - University of Connecticut (Stamford),* *email: nadiakap@optonline.net*

## Introduction

**Optimization lies at the heart of machine learning and data science.**

**One of the most relevant problems in machine learning is automatic selection of optimization algorithm depending on the objective. This is necessary in many applications such as robotics, simulating biological or chemical processes, trading strategies optimization, to name a few.**

**We developed a library of optimization methods as a first step for self-adapting algorithms. Optimization methods in this library work with all objectives including very onerous ones, such as black box functions and functions given by computer code, and the convergence of methods is guaranteed.**

**This library allows to create customized derivative free learning algorithms with desired properties by combining building blocks from this library or other Python libraries.**

**The library is intended primarily for educational purposes and its focus is on transparency of the methods rather than on efficiency of implementation.**

## References


1. Kaplinskij, A.I., Pesin, A.M., Propoj, A.I., Analysis of search methods of optimization based on potential theory. I: Nonlocal properties. Automation and Remote Control, 1994, vol. 9, pp.97-105
2. Kaplinskij, A.I., Propoj, A.I., First-order nonlocal optimization methods that use potential theory, Automation and Remote Control,1994
3. Kaplinskij, A.I., Pesin, A.M., Propoj, A.I., Analysis of search methods of optimization based on potential theory. III: Convergence of methods. Automation and Remote Control, 1994.
4. Nikhil Bansal, Anupam Gupta, Potential-function proofs for gradient methods, Theory of Computing, 2019
5. Adrien Taylor, Francis Bach, Stochastic first-order methods: non-asymptotic and computer-aided analyses via potential functions, Conference on Learning Theory (COLT) 2019
6. Zeyuan Allen-Zhu and Lorenzo Orecchia, Linear Coupling: An Ultimate Unification of Gradient and Mirror Descent, Innovations in Theoretical Computer Science Conference (ITCS), 2017, pp. 3:1-3:22.
7. Berthold Immanuel Schmitt, Convergence Analysis for Particle Swarm Optimization, FAU University Press, 2015, 214 p.
8. Juan Frausto-Solis, Ernesto Linan-Garcia, Juan Paulo Sanchez, Juan Javier Gonzalez Barbosa, Carlos Gonzalez, Guadalupe Castilla-Valdez, Multiphase Simulated Annealing Based on Boltzmann and Bose-Einstein Distribution Applied to Protein Folding Problem, Advances in Bioinformatics, 2016
9. Gong, G., Liu, Y., Qian, M, Simulated annealing with a potential function with discontinuous gradient on Rd,Science in China Series A: Mathematics, 2001,vol. 44, issue 5, pp. 571-578


## Applications

### Robotics

[1]Valdez Peña, Sergio & Hernandez, Eusebio & Keshtkar, Sajjad. (2020). A Hybrid EDA/Nelder-Mead for Concurrent Robot Optimization. 10.1007/978-3-030-14347-3_20. An evolutionary Nelder–Mead slime mould algorithm with random learning for efficient design of photovoltaic models
https://www.sciencedirect.com/science/article/pii/S2352484721011653

### Electrical Engineering

[2]Yi Fan, Pengjun Wang, Ali Asghar Heidari, Huiling Chen, HamzaTurabieh, Majdi Mafarja, Random reselection particle swarm optimization for optimal design of solar photovoltaic modules, Energy,Volume 239, Part A,2022,
https://www.sciencedirect.com/science/article/abs/pii/S0360544221021137

### Chemistry

[3] Fath, Verena, Kockmann, Norbert, Otto, Jürgen, Röder, Thorsten, Self-optimising processes and real-time-optimisation of organic syntheses in a microreactor system using Nelder–Mead and design of experiments, Reaction Chemistry & Engineering, 2020
https://pubs.rsc.org/en/content/articlelanding/2020/re/d0re00081g

### 3D printing

[4] Plüss T, Zimmer F, Hehn T, Murk A. Characterisation and Comparison of Material Parameters of 3D-Printable Absorbing Materials. Materials (Basel). 2022 Feb;15(4) 1503.
https://europepmc.org/article/med/35208040

[5] Thoufeili Taufek, Yupiter H.P. Manurung, Mohd Shahriman Adenan, Syidatul Akma, Hui Leng Choo, Borhen Louhichi, Martin Bednardz, and Izhar Aziz. Modeling and Simulation of Additively Manufactured Cylindrical Component Using Combined Thermomechanical and Inherent Strain Method with Nelder–Mead Optimization, 3D Printing and Additive Manufacturing.
http://doi.org/10.1089/3dp.2021.0197

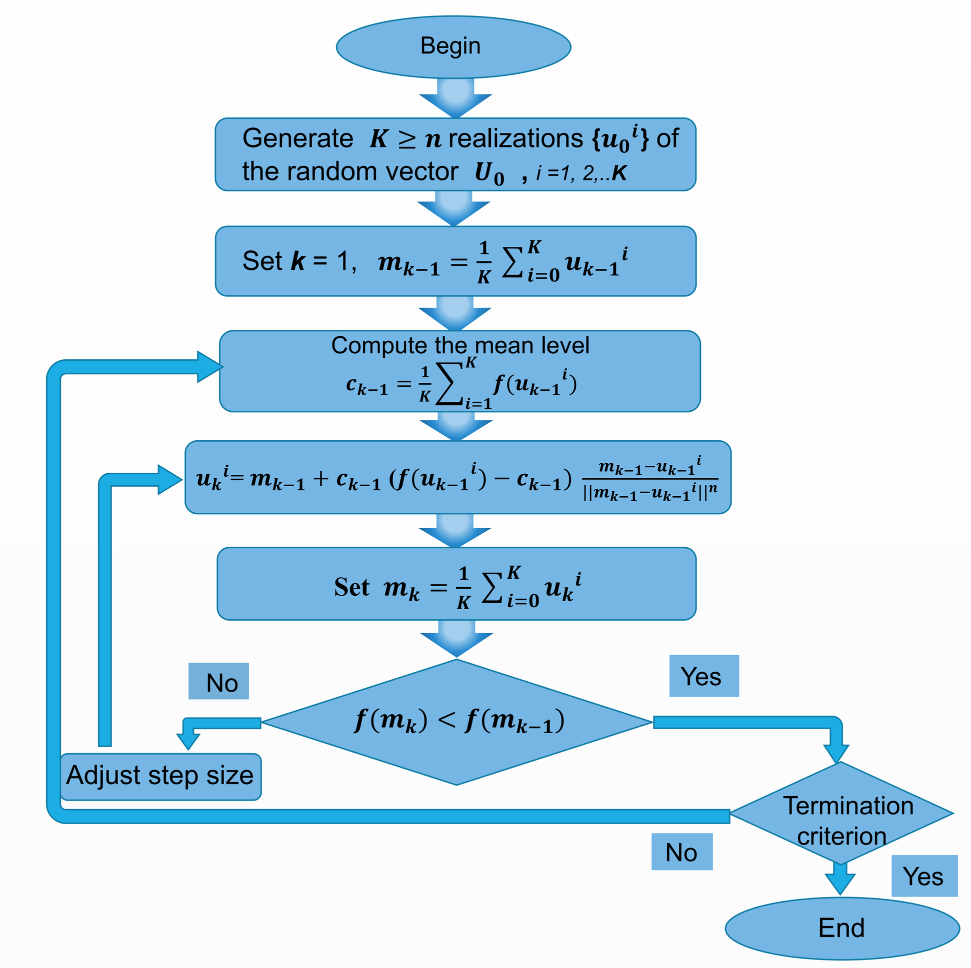
### Wine tasting/production

[6] Vismara, P., Coletta, R. & Trombettoni, G. Constrained global optimization for wine blending. *Constraints* 21, 597–615 (2016).
https://doi.org/10.1007/s10601-015-9235-5

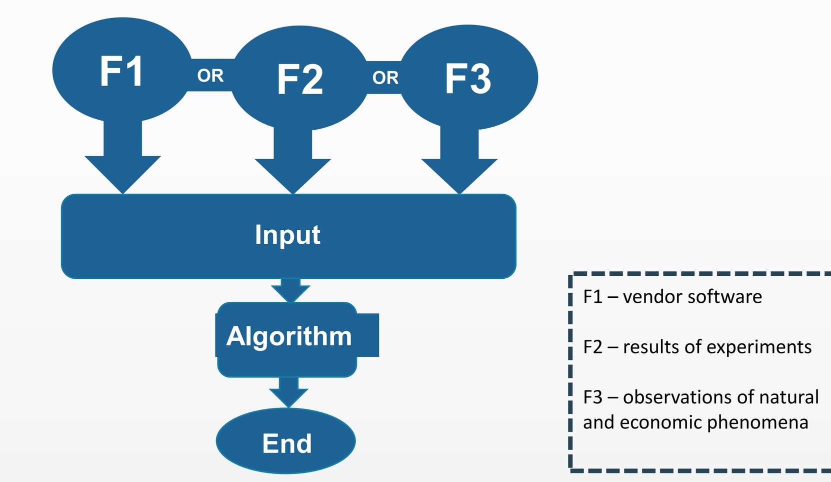
[7] Terry Hui-Ye Chiu, Chienwen Wu, Chun-Hao Chen, A Generalized Wine Quality Prediction Framework by Evolutionary Algorithms, International Journal of Interactive Multimedia and Artificial Intelligence, Vol. 6, Nº7,2021
https://www.ijimai.org/journal/bibcite/reference/2935

- Created for students and researchers
- Has its roots in potential theory
- Optimization methods with desired properties are created based on basic modules, by varying parameters of generalized algorithm
- Allows
  - to design optimization methods with guaranteed convergence but without the use of derivatives of objective function
  - to compare existing optimization methods
  - to understand principles of learning algorithms
  - to design custom variations or hybrids of known heuristic optimization methods.

## Building blocks




## Examples of black box objectives




## *Minpy* code organization

**Minimization**

dim
distribution
initial_guess
n_max_iterations
n_points
objective_function
step_size
tolerance

_contract()
_reflect()
_shrink()
_update_center_of_mass()
_update_mean_level()
_exit_criterion()
minimize_NMExt()
minimize_CMAExt()
minimize_GAExt()
minimize_SAExt()

**TestFunctions**

ackley()
booth()
easom()
himmelblau()
shor()
shor10()
spher()

### Sample assignments for students

1. Create an algorithm based on *minpy* building blocks that performs iteration steps according to generalized algorithm, where initial distribution of points follows normal distribution centered at initial point with unit variance. Keep population size at each iteration unchanged. Use stopping criterion of your choice. Test this algorithm on Ackley function.
2. Create an algorithm based on *minpy* building blocks that performs iteration steps according to generalized algorithm, where initial distribution of points follows exponential distribution with unit variance. Use stopping criterion of your choice. Test this algorithm on Sphere function with 5 arguments.
3. Create an algorithm based on *minpy* building blocks that performs iteration steps according to generalized algorithm, where initial distribution of points follows normal distribution centered at initial point with unit variance, and on each iteration 3 random points with “good” values ( values that are smaller than average function value achieved so far) are added to the population. The algorithm should stop when largest distance between points becomes less than certain small number. Test this algorithm on Booth function.
4. Create an algorithm with a “memory”. On each iteration of the algorithm the new direction is computed as a difference between two previous directions. Algorithm starts from randomly selected simplex with number of points n+1, where n is number of function arguments. Second steps of the algorithm should be computed as reflection of worst initial point from the center of mass of “good” points. Use stopping criterion of your choice. Test on Shor function, consider n = 3 and n = 7 cases.
5. Implement a variant of CMAES using *minpy* building blocks. The algorithm should stop when Kullback – Leibler divergence becomes smaller than some given tolerance.